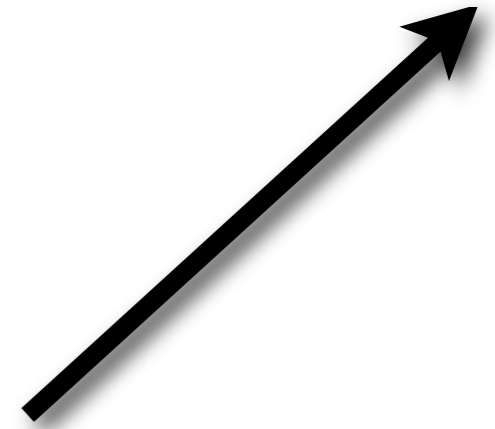


Das sicherste Betriebssystem

Jörg Möllenkamp
Senior Systems Engineer

Sun Microsystems

Bevor jemand fragt:
Ich meine natürlich Solaris ...



Hat sich der geneigte Zuhörer
bei diesem Logo wohl aber
schon gedacht ...

Okay, ich habe folgende Lehre beim Erstellen des Vortrages gezogen ...

Lass niemals einen Vertriebsbeauftragten
den Titel eines technischen Vortrags festlegen ... ;)

Scherz beiseite

Es gibt kein „sicherste Betriebssystem“

Die Sicherheit einer Installation hängt von vielen Faktoren ab, und nur einer ist das Betriebssystem ...

Ein gutes und sicheres Betriebssystem stellt Tools und Mechanismen bereit mit denen ein Administrator sein System *schnell* und *einfach* absichern kann ohne den normalen Nutzer *mehr als nötig* einzuschränken.

Was nutzt das tollste Tool,
wenn man damit kaum umgehen kann?

<sarkasmus>Oder warum steht bei vielen Anleitungen
gleich am Anfang der SELinux-Anleitung,
wie man es wieder ausschaltet ?**</sarkasmus>**

Okay ... zurück zu Solaris ...

Aus Marketingsicht würde man mit der
Common Criteria Evaluierung anfangen

Okay ...

Common Criteria

Ja ... haben wir auch ...

Solaris 10 11/06 ist zertifiziert nach ...

EAL 4+

Role based access control Protection Profile

Conditional Access Protection Profile

Solaris 10 + Trusted Extensions ist in Kürze zertifiziert nach ...

EAL 4+

Role based access control Protection Profile

Conditional Access Protection Profile

Label Security Protection Profile

Interessanterweise wissen nur
wenige was es wirklich mit einer
Common Criteria Zertifizierung
auf sich hat

Sehr böse Zungen behaupten:
Die Zahl nach EAL steht für die Anzahl der Nullen,
die man an 100 anfügen muss, um
die Zertifizierung zu bekommen ... ;)

Am Anfang war der Evaluierungsgegenstand ...

Securityspek für das was getestet werden soll:

Also bei der Common Criteria Zertifizierung von Solaris:

Solaris 10 11/06

Die Nennung
des Evaluation Assurance Levels (EAL)
sagt nichts über die Sicherheit des Systems aus.

Der Evaluation Assurance Level sagt
nur mit welchen Methoden
ein Evaluierungsgegenstand geprüft wird.

- EAL1: Functionally Tested
- EAL2: Structurally Tested
- EAL3: Methodically Tested and Checked
- EAL4: Methodically Designed, Tested and Reviewed
- EAL5: Semiformally Designed and Tested
- EAL6: Semiformally Verified Design and Tested
- EAL7: Formally Verified Design and Tested

Was muss man also mehr wissen?

Man muss wissen,
welche gegen welche Protection Profiles überprüft wurden !

Ein Protection Profile ist
eine Menge von vordefinierten Anforderungen,
gegen die ein Evaluierungsgegenstand
geprüft wird.

Die Nennung der
Common Criteria Einstufung
und der
Protection Profiles
sagt auch nichts über die Sicherheit des Systems aus!

Man muss das das Security Target kennen
genau kennen !

Das Security Target beschreibt, was mit einem Evaluierungsgegenstand gemacht wird um die Anforderungen des Protection Profiles zu erfüllen.

Überprüft man all das, kann herauskommen,
das eine EAL Evaluierung nicht ganz so toll ist,
zunächst gedacht ...

Schönes „schlechtes“ Beispiel:

Die EAL4+ Zertifizierung eines
Markbegleiters im Bereich
der unixoiden Betriebssysteme ...

Es wurde unter anderem auch die
Clientvariante getestet ...

Aus der Security Target Definition der Zertifizierung:

- The Client product includes additional packages that are not subject to this evaluation and have been excluded from the evaluated configuration. In the evaluated configuration, both the Client and Server products are configured for “server” use, without a graphical desktop.

Aus dem Evaluation Report der Zertifizierung:

10. VALIDATOR COMMENTS

The Validator determined that the evaluation and all of its activities were performed in accordance with the CC, the CEM and CCEVS practices.

The Validator has the following observation:

- While the TOE distribution media includes a Graphical User Interface (GUI), it is not installed by default, is not part of the Evaluated Configuration and was not evaluated.

Ein Client ohne graphischen Desktop ?

Meine Empfehlung daher:

Sehr genau die Zertifizierungsunterlagen
lesen!

Schönes „sehr schlechtes“ Beispiel:

Die Zertifizierung für Windows NT
vor vielen Jahren ... die nur galt, wenn
das System nicht an Netzwerke
angeschlossen war.

Bei Solaris beinhaltet das Security von den ACLs ueber RBAC bis hin zu Userdatenbanken im LDAP inclusive eingeschalteten Desktop eine Vielzahl von Features.

Bei den Trusted Solaris bzw. den Trusted Extension berücksichtigt die Evaluierung sogar die Berücksichtigung von Geheimhaltungsstufen beim Drucken, bei Übertragung von Daten via NFS....

... sogar beim Cut n' Paste zwischen Fenstern mit Daten unterschiedlicher Geheimhaltung.

Für Solaris 10 11/06:

Security Target:

<http://www.sun.com/software/security/securitycert/docs/solaris10-sec-e.pdf>

Certification Report:

<http://www.sun.com/software/security/securitycert/docs/solaris10-cert-e.pdf>

Und mal ganz ehrlich

Eine erhaltene
Certification sagt an sich auch noch wenig ueber
die Sicherheit eines Systems aus ...

Viele Sicherheitsprobleme haben
ihren Ursprung vor der Tastatur.

Security features in Solaris

Daher werde ich in meinem Vortrag
mich eher auf praktische Dinge verlagern ...

Features, die es erleichtern ein System
sicherer zu machen...

Ich kann in 30 Minuten nur eine
kleine Auswahl streifen ...

Secure by default

Solaris hat den Ruf, in der Standard
Konfiguration zuviele Dienste im
Netz zur Verfügung zu stellen ...

„Portscan“ wurde gestartet ...

Port Scanning host: 10.211.55.3

Open TCP Port:	21	ftp
Open TCP Port:	22	ssh
Open TCP Port:	23	telnet
Open TCP Port:	25	smtp
Open TCP Port:	79	finger
Open TCP Port:	111	sunrpc
Open TCP Port:	513	login
Open TCP Port:	514	shell
Open TCP Port:	587	submission
Open TCP Port:	898	

„Portscan“ wurde beendet ...

Das dem so ist, hängt mit der
Binärkompatibilitätsgarantie
von Solaris zusammen ...

Ein Programm könnte darauf
angewiesen sein, das der Dienst
genauso wie in alten Versionen offen
verfügbar ist.

Secure by Default bedeutet:
Vom Netz aus ist nur der
SSH-Port erreichbar.

Will man den Zustand vor „Secure by default“

```
# netservices open
```

Und so macht man das System wieder zu ...

```
# netservices limited
```

Secure by Default ist heute
voreingestellt bei der Installation.

Der Auslieferungszustand sieht heute so aus ...

„Portscan“ wurde gestartet ...

Port Scanning host: 10.211.55.3

Open TCP Port: 22 ssh

„Portscan“ wurde beendet ...

Die meisten Dienste sind weiterhin aktiv ...

Sie horchen nur nicht mehr
auf externe
Netzwerkinterfaces...

RBAC

Das herkömmliche Rechtemodell
ist zweigeteilt ...

Der normale User:
kann wenig

root:
kann alles

Herausforderung:
Manche Dinge in einem Unix-Rechner erfordern
root-rechte

Problem:
Will man jedem Admin wirklich das Password
für den Root-Account geben ?

Ein Admin, der eigentlich nur
für das Hinzufügen von Druckern
verantwortlich ist, kann auch das
Auditing ausschalten
am Cluster rumspielen
Netzwerkinterfaces umkonfigurieren

Dieses Problem ist so alt
wie Unix selbst.

1980 wurde daher sudo erfunden.

sudo steht übrigens nicht für
super user do

sudo steht für
substitute user do

Solaris löst das Problem etwas anders ...

RBAC=Role based access control

Als Admin hat man immer eine Rolle.

Man ist normaler User, aber auch:
Printer Admin
Operator
User Admin
Filesystem Admin

Ein Rolle hat immer ein Rollenprofil ...

Das Rollenprofil legt fest:

- welche Programme die Rolle ausführen kann
- welche Authorisations eine Rolle hat

Eine oder mehrere Rollenprofile
werden einer Rolle zugewiesen.

Eine Rolle ist technisch eine Nutzeraccount
in den man sich nicht direkt einloggen kann.

Man loggt sich als normaler Nutzer ein ...

... und nimmt eine Rolle an.

Nur so lange man diese Rolle angenommen hat,
hat man die Sonderrechte dieser Rolle.

Hat man seine Arbeit erledigt,
dann verlässt man die Rolle und
ist wieder normaler User.

Bisher klingt das ganze
nach einem etwas erweiterten
sudo.

Sun RBAC kennt zusätzlich
das Konzept
der Authorizations.

Authorisations löst folgende Frage:
Wie kann man einem Nutzer
nur erlauben nur bestimmte Funktionen
eines Programms auszuführen?

Authorisations benötigen die Unterstützung der Applikationen.

Allerdings unterstützen viele Programme
des Solaris Operating Environments
Authorisations.

Der Kernel stellt nur Funktionen bereit,
um einer Applikation mitzuteilen,
welche Authorisations eine Rolle hat.

Auch in einem frisch installierten Solaris
werden Authorisations genutzt.

Jeder Nutzer hat die Authorisation:
solaris.device.cdrw.*

Dadurch kann jeder Nutzer CD lesen und beschreiben ...

Stellen Sie sich das wie ein Werkzeugschrank
mit einer Bohrschraubmaschine vor.

Die Rolle ist der Schlüssel
zum Werkzeugschrank

Mit der Authorisation überprüft der Bohrschrauber
ob der Handwerker authorisiert ist:

- ein Loch zu bohren
- Schrauben aus der Wand zu drehen
- den Bohrer oder das Bit zu wechseln

Wozu man das braucht ?

Wenn Sie verhindern wollen, dass jemand wieder die Stromleitung anbohrt, aber sie der Person schon erlauben wollen, Schrauben aus der Wand zu holen ...

Um auf Solaris zurück zu kommen ...

Sie wollen möglicherweise nicht jedem Admin jede Funktion eines Tools zugänglich machen.

Least Privileges

There is no root!

Okay,
es gibt noch einen User root
mit der UserID 0
der alle Rechte eines root hat ...

Ja hängt auch wieder mit der
Binärkompatibilitätsgarantie
zusammen ...

Sie können auch jedem anderen Nutzer
root-gleiche Rechte geben ...

Aber das muss so nicht mehr sein ...

Es ist nur noch aus Kompatibilitätsgründen so!

Solaris 10 arbeitet mit Privilegien

Frage in die Runde:
Warum muss Apache mit root-rechten
gestartet werden?

Richtig:
Port 80 ist ein privilegierter Port ...

Man brauchte bisher root-Rechte, um dieses Privileg zu erlangen...

There is no root!

Ich kann unter Solaris 10
das Privileg an User vergeben,
sich an privilegierte Ports zu binden!

Die Privilegien eines Root-Users

contract_event contract_observer cpc_cpu dtrace_kernel
 dtrace_proc dtrace_user file_chown file_chown_self
 file_dac_execute file_dac_read file_dac_search file_dac_write
 file_downgrade_sl file_flag_set file_link_any file_owner file_setid
 file_upgrade_sl graphics_access graphics_map ipc_dac_read
 ipc_dac_write ipc_owner net_bindmlp net_icmpaccess
 net_mac_aware net_privaddr net_rawaccess proc_audit
 proc_chroot proc_clock_highres proc_exec proc_fork proc_info
 proc_lock_memory proc_owner proc_prioctl proc_session
 proc_setid proc_taskid proc_zone sys_acct sys_admin sys_audit
 sys_config sys_devices sys_ip_config sys_ipc_config sys_linkdir
 sys_mount sys_net_config sys_nfs sys_res_config sys_resource
 sys_smb sys_suser_compat sys_time sys_trans_label
 win_colormap win_config win_dac_read win_dac_write
 win_devices win_dga win_downgrade_sl win_fontpath
 win_mac_read win_mac_write win_selection win_upgrade_sl

Rechte eines normalen Users ...

contract_event contract_observer cpc_cpu dtrace_kernel
 dtrace_proc dtrace_user file_chown file_chown_self
 file_dac_execute file_dac_read file_dac_search file_dac_write
 file_downgrade_sl file_flag_set **file_link_any** file_owner file_setid
 file_upgrade_sl graphics_access graphics_map ipc_dac_read
 ipc_dac_write ipc_owner net_bindmlp net_icmpaccess
 net_mac_aware net_privaddr net_rawaccess proc_audit
 proc_chroot proc_clock_highres **proc_exec** **proc_fork** **proc_info**
 proc_lock_memory proc_owner proc_prioctl **proc_session**
 proc_setid proc_taskid proc_zone sys_acct sys_admin sys_audit
 sys_config sys_devices sys_ip_config sys_ipc_config sys_linkdir
 sys_mount sys_net_config sys_nfs sys_res_config sys_resource
 sys_smb sys_suser_compat sys_time sys_trans_label
 win_colormap win_config win_dac_read win_dac_write
 win_devices win_dga win_downgrade_sl win_fontpath
 win_mac_read win_mac_write win_selection win_upgrade_sl

Um noch mal auf das Beispiel
des Apache-Servers zurueck zu
kommen ...

Was braucht jetzt ein Apache Server

contract_event contract_observer cpc_cpu dtrace_kernel
 dtrace_proc dtrace_user file_chown file_chown_self
 file_dac_execute file_dac_read file_dac_search file_dac_write
 file_downgrade_sl file_flag_set **file_link_any** file_owner file_setid
 file_upgrade_sl graphics_access graphics_map ipc_dac_read
 ipc_dac_write ipc_owner net_bindmlp net_icmpaccess
 net_mac_aware **net_privaddr** net_rawaccess proc_audit
 proc_chroot proc_clock_highres **proc_exec** **proc_fork** **proc_info**
 proc_lock_memory proc_owner proc_prioctl **proc_session**
 proc_setid proc_taskid proc_zone sys_acct sys_admin sys_audit
 sys_config sys_devices sys_ip_config sys_ipc_config sys_linkdir
 sys_mount sys_net_config sys_nfs sys_res_config sys_resource
 sys_smb sys_suser_compat sys_time sys_trans_label
 win_colormap win_config win_dac_read win_dac_write
 win_devices win_dga win_downgrade_sl win_fontpath
 win_mac_read win_mac_write win_selection win_upgrade_sl

Was braucht jetzt ein Apache Server

contract_event contract_observer cpc_cpu dtrace_kernel
 dtrace_proc dtrace_user file_chown file_chown_self
 file_dac_execute file_dac_read file_dac_search file_dac_write
 file_downgrade_sl file_flag_set **file_link_any** file_owner file_setid
 file_upgrade_sl graphics_access graphics_map ipc_dac_read
 ipc_dac_write ipc_owner net_bindmlp net_icmpaccess
 net_mac_aware **net_privaddr** net_rawaccess proc_audit
 proc_chroot proc_clock_highres **proc_exec** **proc_fork** proc_info
 proc_lock_memory proc_owner proc_prioctl **proc_session**
 proc_setid proc_taskid proc_zone sys_acct sys_admin sys_audit
 sys_config sys_devices sys_ip_config sys_ipc_config sys_linkdir
 sys_mount sys_net_config sys_nfs sys_res_config sys_resource
 sys_smb sys_suser_compat sys_time sys_trans_label
 win_colormap win_config win_dac_read win_dac_write
 win_devices win_dga win_downgrade_sl win_fontpath
 win_mac_read win_mac_write win_selection win_upgrade_sl

Was braucht jetzt ein Apache Server

contract_event contract_observer cpc_cpu dtrace_kernel
 dtrace_proc dtrace_user file_chown file_chown_self
 file_dac_execute file_dac_read file_dac_search file_dac_write
 file_downgrade_sl file_flag_set **file_link_any** file_owner file_setid
 file_upgrade_sl graphics_access graphics_map ipc_dac_read
 ipc_dac_write ipc_owner net_bindmlp net_icmpaccess
 net_mac_aware **net_privaddr** net_rawaccess proc_audit
 proc_chroot proc_clock_highres **proc_exec** **proc_fork** proc_info
 proc_lock_memory proc_owner proc_prioctl proc_session
 proc_setid proc_taskid proc_zone sys_acct sys_admin sys_audit
 sys_config sys_devices sys_ip_config sys_ipc_config sys_linkdir
 sys_mount sys_net_config sys_nfs sys_res_config sys_resource
 sys_smb sys_suser_compat sys_time sys_trans_label
 win_colormap win_config win_dac_read win_dac_write
 win_devices win_dga win_downgrade_sl win_fontpath
 win_mac_read win_mac_write win_selection win_upgrade_sl

Und das haben sie bisher vergeben ...

contract_event contract_observer cpc_cpu dtrace_kernel
 dtrace_proc dtrace_user file_chown file_chown_self
 file_dac_execute file_dac_read file_dac_search file_dac_write
 file_downgrade_sl file_flag_set file_link_any file_owner file_setid
 file_upgrade_sl graphics_access graphics_map ipc_dac_read
 ipc_dac_write ipc_owner net_bindmlp net_icmpaccess
 net_mac_aware net_privaddr net_rawaccess proc_audit
 proc_chroot proc_clock_highres proc_exec proc_fork proc_info
 proc_lock_memory proc_owner proc_prioctl proc_session
 proc_setid proc_taskid proc_zone sys_acct sys_admin sys_audit
 sys_config sys_devices sys_ip_config sys_ipc_config sys_linkdir
 sys_mount sys_net_config sys_nfs sys_res_config sys_resource
 sys_smb sys_suser_compat sys_time sys_trans_label
 win_colormap win_config win_dac_read win_dac_write
 win_devices win_dga win_downgrade_sl win_fontpath
 win_mac_read win_mac_write win_selection win_upgrade_sl

Weitere interessante Features ...

Auditing

Die Fragestellung:
Was passiert auf meinem System?
Wann hat wer welche Kommandos ausgeführt?

Auditing

Solaris Auditing beantwortet diese Fragestellung

```
header,124,2,AUE_EXECVE,,localhost,2008-02-02 00:12:49.560 +01:00  
path,/usr/bin/ls  
attribute,100555,root,bin,26738688,1380,0  
exec_args,2,ls,-l  
subject,jmoekamp,root,root,root,root,665,2040289354,12921 71168 10.211.55.2  
return,success,0
```

Auditing

Mit Solaris Auditing kann man eine Vielzahl von Aktionen im System überwachen ...

Basic Audit Reporting Tool

Fragestellung:

Ist das System wirklich noch so wie ich es installiert habe ?

Oder die Standardausrede eines jeden Admin¹
wenn was schief geht:

„Ich habe wirklich nichts geändert!!!“

¹ Ich eingeschlossen ...

Basic Audit Reporting Tool

Nach der Installation lässt man das BART ein erstes Mal über die Dateisysteme laufen.

Basic Audit Reporting Tool

Damit hat man eine Baseline für spätere
Vergleiche ...

Diese Datei gut weglegen!

Basic Audit Reporting Tool

Der Tag ist da ... man will wissen, was an einem System verändert worden ist.

Basic Audit Reporting Tool

Man lässt nochmals das BART Tool über die Installation laufen.

Basic Audit Reporting Tool

Und dann vergleicht man diese Versionen.

Basic Audit Reporting Tool

```

/nsswitch.files:
mode control:100644 test:100777
acl control:user::rw-,group::r--,mask:r--,other:r--
test:user::rwx,group::rwx,mask:rwx,other:rwx
/nsswitch.nisplus:
size control:2525 test:2538
mtime control:473976b5 test:47a44862
contents
control:79e8fd689a5221d1cd059e5077da71b8
test:3f79176ec352441db11ec8a3d02ef67c
/thisisjustatest:add

```

Das kommt dabei raus, wenn man zwischen den beiden Starts von BART ...

1. die rechte an /etc/nsswitch.files ändert,
2. an /etc/nsswitch.nisplus eine Zeile anhängt,
3. und ein touch /etc/thisisjustatest ausführt.

Signed Binaries

Okay, ich habe hier die `/usr/sbin/ifconfig`
... aber ist die Datei auch von Sun?

Signed Binaries

Jedes Programm des Solaris Operating Environment ist von Sun digital signiert.

Signed Binaries

```
# elfsign verify -v /usr/sbin/ifconfig
elfsign: verification of /usr/sbin/ifconfig passed.
format: rsa_md5_sha1.
signer: CN=SunOS 5.10, OU=Solaris Signed Execution, O=Sun
Microsystems Inc.
```

Wollen Sie mehr wissen?

www.c0t0d0s0.org/pages/lksf_content.html#nubit